

DOCUMENT RESUME

ED 259 558

FL 015 118

AUTHOR Edwards, Jane A.
TITLE Two Applications of Computers to Second Language Research.
PUB DATE Apr 85
NOTE 11p.; "Transcript Analysis" is the newsletter of the Child Language Data Exchange System at Carnegie-Mellon University.
PUB TYPE Reports - Evaluative/Feasibility (142) -- Journal Articles (080)
JOURNAL CIT Transcript Analysis; v2 n1 p7-16 Apr 1985
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Adults; Child Language; *Comparative Analysis; *Data Processing; Immigrants; *Information Storage; Information Utilization; Language Research; *Programming Languages; *Second Languages; Standards; *Structural Analysis (Linguistics)

ABSTRACT

The results of an examination of computer data formatting and data analysis strategies from two diverse second language projects and their implications for establishing standards are outlined in the form of suggested formatting conventions and analysis procedures. The formatting suggestions address the issues of distinguishing line types, avoidance of needless variation and random error in typing, the scope of comments and codes, coding, and sub-units or analytic subdivisions within the text. The analysis strategies focus on data selection, data organization to reveal regularities, and data modification through computer-assisted coding. The importance of selecting formatting conventions that are theoretically flexible, computationally flexible, readable by the human eye, quick to type onto the computer, robust against typing error, and expandable is emphasized, and the establishment of a minimal set of conventions to serve as an international standard and to facilitate collaboration and data exchange across projects, including comparisons with first language acquisition research, is supported. (MSE)

* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *

3. Two Applications of Computers to
Second Language Research"PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BYBrian
McWhinneyTO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)."U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)☐ This document has been reproduced as
received from the person or organization
originating it.☒ Minor changes have been made to improve
reproduction quality.☐ Points of view or opinions stated in this docu-
ment do not necessarily represent official NIE
position or policy.Jane A. Edwards
Institute of Cognitive Studies
University of California
Berkeley, California 94720

and

Max-Planck-Institut für Psycholinguistik
Berg en Dalseweg 79
NL-6522 BC Nijmegen
The Netherlands

BEST COPY

3.1 Abstract

The author describes computer procedures (data formatting and data analysis strategies) from two diverse second-language projects: the study of adult immigrants in France, Germany, England, Sweden, and Holland sponsored by the European Science Foundation, and the study of individual differences in English acquisition among Chicano and Cantonese school children directed by Lily Wong Fillmore. The author notes the importance of selecting formatting conventions which are theoretically flexible, computationally flexible, readable by the human eye, quick to type onto computer, robust against typing error, and expandable. The paper supports the establishment of a minimal set of conventions which could serve as an international standard, and which could ease collaboration and data exchange across projects, including comparisons with first language acquisition research.

3.2 Introduction

Acquisition research using large natural language texts involves several distinct activities. The researcher (a) locates instances of a particular linguistic type (e.g., first person pronouns, verbal complements, adverbs, inherently perfective verbs, past tense verb forms, etc.), (b) identifies regularities across these instances, (c) hypothesizes reasons explaining observed regularities, and (d) tests these hypotheses by means of additional data. While human intelligence is necessary for the scientific insights and intuitive leaps involved in selecting phenomena of importance and in interpreting observed regularities, the time-consuming clerical operations that are part of the research process can be greatly speeded and accomplished more accurately by computer. These operations include: (a) locating patterns within a context of user-specified size (the "data selection function"), (b) displaying or ordering the data in such a way that patterns are more visually prominent to the

ED259558

81151078

human eye. (the "data organization" function), and (c) easing the marking of particular distinctions within the data, i.e., computer-assisted coding of the data (the "data modification" function). Computer processes such as these can be effectively employed even on largely uncoded data and even at the early stages of data analysis. More critical for their effective use than a fully coded data set is the adoption of resourceful and flexible strategies in applying them, and an appropriately formatted data base.

Principles underlying data formatting and analysis strategies constitute the focus of this paper. The *formatting conventions* recommended here are in large part borrowed from either or both of two second language projects: the European Science Foundation sponsored study of acquisition by adult migrant workers in England, France, Germany, Holland, and Sweden (the "ESF project"); and Lily Wong Fillmore's study of individual differences in acquiring English as a second language among Chinese children and Chicano children (the "Individual Differences project"). In addition, they bear a close formal resemblance to the conventions being adopted by the international child language data base (Child Language Data Exchange System) being established by Brian MacWhinney and Catherine Snow (*Journal of Child Language*, in press). The *analysis procedures* mentioned here are of similar generality, and are widely available. They exist (under various names) on most large computer systems, and are in the process of being programmed in C and PASCAL by CHILDES. What is emphasized here is the manner in which simple tools can be used in powerful ways to address diverse research questions.

3.3 Formatting conventions for computer-processed data files.

The level of detail found in the script of a play is a suitable level of detail for explicating the formatting conventions appropriate for computer files. When properly designed (as below) the computer transcription can be made increasingly detailed without limit as suits the needs of the researcher and the research question (i.e., it is flexibly expandable), without requiring compensatory adjustments of the analysis tools. A transcript at this most basic level of detail preserves what are generally regarded as the most essential aspects of the data gathering session: *a full record of verbalized events (including speech repairs, overlaps, hesitations, and interruptions, with adequate contextual information, some indication of intonational and pausal aspects, and basic gestural information.)* The basic format described here consists of two "tiers" (i.e., qualitatively different types of information): utterances and comments. More detailed transcripts contain either more tiers or greater specification of individual tiers (e.g., precise temporal information)--or both. The data format developed in Lily Fillmore's Individual Differences project at Berkeley (described in section 2.2) has extra tiers, for morphological, semantic/syntactic, and function coding (a separate line for each). The data format developed by Lois Bloom at Columbia University includes descriptions of gestures for speaker as well as hearer(s), along with precise temporal information, for studying simultaneity and

exact interleaving of verbal and nonverbal behaviors of all interactants. This level of detail exceeds that usually sought. Where this level of detail is sought, however, the researcher should know about Bloom's specially-designed computer-assisted transcription system which runs on an Apple computer and enables temporal precision in the most efficient and convenient way possible. In any case, the format presented here is flexibly expandable into higher levels of detail should this be desired either at the start or later on.

In inputting transcript information onto the computer, the researcher opens a data file (labelled by an appropriately mnemonic filename) and begins data entry by typing a "header" which serves a documentational and orienting function, detailing features of the data gathering situation, the interactants, etc., and mentioning (by "keywords") particular research questions for which the transcript might be especially useful.

For the actual text the selected formatting conventions must be: *theoretically flexible, computationally flexible, readable by the human eye, quick to type onto computer, robust against typing error and expandable at any time without requiring reorganization of the system as a whole.* The set of minimally-constraining formatting conventions recommended here satisfies these criteria. Of greatest importance are the following five design features:

1. The means used for distinguishing comments from utterances, and utterances of one speaker from utterances of another.
2. The avoidance of needless variation and random error in typing.
3. The "scope" of comments and codes.
4. Coding.
5. Sub-units or analytic subdivisions within the text.

3.3.1 Distinguishing Line Types

The traditional convention in type-written transcripts of indicating speakership by column (e.g., speaker A in the left column, speaker B in the right column) is undesirable in a computer data format because of potential ambiguities for whose resolution a computer is less well-suited than the human eye. For example, if by sloppy typing the left speaker column extended too far to the right, or the right speaker column began too far to the left, individual words or entire utterances might be attributed to the wrong speaker, unless the researcher invested substantial time in the development of a special-purpose program for interpreting these and other format ambiguities.

A more secure and efficient strategy is the use of "line-initial identifiers", that is, the explicit

labelling of each line (even continuation lines, by means of identifiers placed in a fixed column location before the text of each line.) For utterance lines, the identifier may be a speaker initial; for comments, a dash (-); and so on, for any other line-type (or "tier") in the text (e.g., videotape meter readings, codes, foreign language gloss, etc.). The use of line-initial identifiers enables direct line selection (e.g., learner's lines) in a highly efficient and error-free way, minimizing the possibility of attributing text to the wrong speaker, or of misinterpreting comment as text, and, furthermore, it is faster to type than the traditional columnar format. (If preferred for reasons of readability, a columnar format can be generated by a simple reformatting of the input file, as described below, but the line-initial identifiers should be preserved also on the reformatted version.)

3.3.2 Avoidance of needless variation and random error in typing

For minimum error and maximum ease and speed of typing, the number of arbitrary formatting conventions (which disrupt attention to the words themselves) should be minimized, and all needless variation (which may complicate later searches or require a lot of extra editing) should be avoided. There are eight general rules of thumb.

1. **Use only lower case, standard ASCII letters for utterances and comments.** (e.g., "i" instead of "I" as first person singular pronoun, and no capital letters at the beginning of utterances). Upper case may be typed in for phonetics or coding. For upper/lower case distinction between learner and non-learner utterances, the utterances should initially be entered all in lower case and then (still preserving line-initial identifiers) modified by a short filter program which converts all letters following a particular line identifier to upper case.
2. **Avoid non-printing characters, control characters, and text formatting commands in data files.** The visual representation of data should look the same on the screen as on the printed output. When present, control characters and text formatting commands clutter the screen, interfering with rapid scanning of data. For phonetics or for languages other than English, it is preferable to employ combinations or novel uses of the standard ASCII printing characters (e.g. Capital S for /s/) rather than programming special characters. Specially-programmed characters are undesirable for reasons of portability and the necessity of special treatment in analysis programs. Finally, the pagebreak and word wrap characters of Word Star or other small computer editors should be carefully avoided by entering data in these editors exclusively in the "non-document" mode. Word wrap characters convert word-final characters to italics, greatly complicating pattern search procedures.
3. **Keep all lines shorter than the width of the computer screen.** This, again, is for ease of data scanning and comparable formats on the screen as on paper.
4. **Do not hand-number utterances or other units of any size.** Hand-numbering is unnecessary, error-prone, and time-consuming. It can be done more easily, more quickly, and more adequately by computer (see below).

5. **Use one space between words and before punctuation marks; avoid tab characters.** Tab characters are visually indistinguishable from spaces but have a different ASCII representation and hence could cause misleading results in pattern searches in which spaces are sometimes used for more unique pattern specification (e.g., " his " instead of "his", as a means of avoiding retrieval also of such items as "this", or "history").
6. **Reserve punctuation marks for special functions.** Examples might be period (".") for falling intonation and question mark ("?",) for rising intonation. Avoid using any punctuation which has not been explicitly defined for some specific transcription function, as these will introduce unexpected and undesirable variation in utterances.
7. **Normalize commonly occurring expressions or forms where exact surface form is inconsequential.** An example of this is the use of "mhm", for indicating back-channel behavior even though the actual surface form was "hmm". The original form may be preserved along with the normalization.
8. **Avoid disruption of lexical items by special symbols, unless non-disrupted version is also provided.** If a speaker uses faulty syllable stress, or a unique tone contour, which should be captured in the transcript, the item should be provided also in non-disrupted form, as an aid to pattern-search procedures.

3.3.3 The scope of comments and codes

Some comments or codes concern a single word (e.g., a modal verb) or location (e.g., a pause) in an utterance; others concern strings of several words. The former have a scope of 1 element; the latter, scope greater than 1. In the latter case, it is useful to bracket the referred-to elements on the utterance line by means of paired symbols such as parentheses, curly brackets or angle brackets, and then to place the comment or code on the immediately-following line, preceded by an appropriate line-initial identifier (e.g., a dash (-) for comments). The following example is drawn from the ESF conventions. The letters "a" and "b" at the beginning of lines in these examples indicate speakers.

a string1 <string2 string3>

- <stressed>

a string4

² Scope may also arise in conjunction with overlap among utterances such as with this example (ESF conventions):

a string1 [string2 string3 string4]

b [string1 string2]

a string5

² Editorial Note: In the notation recommended by MacWhinney and Snow, this would appear as:

A string1 <string2 string3> [stressed] string4

Conversion between the ESF format and the CHILDES format can be done mechanically, since both are unambiguous.

where string1 and string2 from speaker b overlapped with the central three strings of an utterance by speaker a.³ Where the comment or code concerns an entire utterance, or where it is unnecessary (because obvious to the reader) to indicate the precise items to which the comment or code refers, the comment or code line may be given without bracketing being necessary on the utterance line.

3.3.4 Coding

For practical reasons coding must often be kept to a minimum. The analytic strategies to be discussed below can be applied quite effectively to non-coded data. Coding, can, however, be a powerful tool when properly done, and can greatly facilitate the "data organization" process. The choice of level of detail in coding will vary from project to project also as a function of the research question being investigated. The conventions of the ESF project, for example, specify no additional coding. In contrast, the Individual Differences Project has employed an extremely elaborate coding system. Many projects will find themselves in-between these two extremes.

Because of its thoroughness and excellent design, anyone intending to add codes of any type to their data should first consider the distinctions made in the Individual Differences coding manual since large parts of it may well be directly applicable. Developed over a period of three years by Lily Fillmore, Ann Peters, Jan-Ola Ostman, Thomas Larsen, and Mary Catherine O'Connor, the Individual Differences Coding system provides clearly operationalized, mutually exclusive and seemingly exhaustive set of categories for comprehensive coding of phenomena at morphological, syntactic, and functional levels. The precise nature and functioning of this code is described in greater detail along with the analysis strategies (below).

3.3.5 Sub-units or analytic subdivisions within the text

Depending on the research question, any of a number of potential unit sizes may be appropriate (clause, utterance, turn, adjacency pair, speech act, conversation topic). The Individual Differences Project adopted a somewhat syntactically-motivated unit; the ESF project chose a unit which is related to a "turn" in discourse but which considers overlapping utterances or momentary interruptions (where the first speaker still retains the floor) as still within the same unit, or "group".

³Editorial Note: In the notation recommended by MacWhinney and Snow, this would appear as:

A: string1 <string2 string3 string4>[overlap below]

B: <string1 string2>[overlap above]

Here, again, the conversion between the two notations can be handled mechanically.

Regardless of which of many possible units is selected, data analysis is eased by assigning a common unit identification number to the various tiers within a unit. Unit numbers should, however, never be added by human typists. They can easily be generated by computer, merely by giving a systematic cue for the beginning of each new unit. In the Individual Differences Project, the cue is any line-initial identifier which is a capital letter (a tacit cue). In the ESF project the cue is an arbitrary symbol (#) placed on the left margin in-between adjacent units (an explicit record separator):

```
#
x here is some sample text from speaker x
x which continues for two lines
#
q now speaker q says something
```

In either case, it is a simple matter to write a program which increments the unit identification number whenever it detects a unit cue on the left margin. For both data sets the result is something like the following:

```
055 x: here is some sample text from speaker x
055 x: which continues for two lines
056 q: now speaker q says something
```

In addition to numbering unnumbered units, the computer can be useful in re-numbering previously-numbered units, in the event of additional unit subdivision.

Programs can modify other aspects of the input format as well, such as the added colon and an extra space following the speaker initial in the above example, or the multi-column spatial display of utterances by difference speakers (though note the preservation of the line identifiers in addition to the column structure, so as to ensure maximal ease in data analysis):

```
#
055 x: here is some sample text from speaker x
055 x: which continues for two lines
#
056 q. now speaker q says something
#
[etc.]
```

Such reformatting programs are currently being programmed in C and PASCAL at CHILDES. A similar multi-column effect could be achieved, though somewhat more crudely, through a global substitution command within the user's own text editor, replacing, for example, line-initial "q" with "q:" (once more preserving q, the line identifier).

3.4 Analysis strategies for computer-processed data files.

3.4.1 Data selection

On most large computers there exist pattern search utilities as part of the operating system which will retrieve the particular line on which a user-specific pattern occurs in a file. They usually permit spaces in the pattern specification (e.g., "his " rather than "his") and many may have wildcard character, enabling the search for discontinuous items (e.g., "put*down" as a means for locating both "put the book down" and "put down the book"). Rather than only retrieving the line on which the pattern occurred, it is generally of far greater use in language acquisition research to retrieve also several lines of context. The context surrounding the pattern-bearing line can be either an arbitrary (user-specified) number of lines, or a logical chunk of text (e.g., bounded by record separators, as in the ESF format, or sharing the same unit identification number, as in the Individual Differences format). While these are rarer as standard system utilities, such programs can be easily devised, and thereby can absorb the burden of what is probably *the single most frequent operation performed by hand with non-computerized acquisition data sets.*

3.4.2 Data organization: Revealing regularities in data

A type/token frequency analysis of a learner's utterances gives a general inventory of occurring forms. From this type of listing, the researcher can identify immediately, for example, which particular personal pronouns occurred in the text and which ones were conspicuously absent, and how frequently they all occurred relative to one another. Beginning with this quick insight, the researcher could then proceed to a closer examination of the syntactic, semantic, and discourse functions of these forms.

Another common tool is a concordance, which gives a restricted (usually one-line) context for every word in the text. Together with the type/token frequency listing, this tool might be used, in studying, for example, the elaboration of the noun phrase during acquisition. The researcher can immediately scan the concordance entries for definite and indefinite articles. The type/token frequency listing, which serves as a quick index of the occurring forms in the text, can then be scanned for additional elements related to noun phrase development (e.g., numbers, demonstratives), which can then be examined in phrasal context on the concordance listing.

A third, and often overlooked tool, is the alphabetic sort. In many cases, the researchers may wish quickly to see the manner in which two sets of qualitatively different phenomena covary. For example, the researcher might be interested in knowing in what ways the syntactic form of an utterance (imperative, question, etc.) predicts its pragmatic function (request, etc.). One can imagine a scenario in which the researcher copies his original data file to a "scratch file" for purposes of

experimentation and analysis. This researcher then proceeds through this file, inserting at the beginning of each of line spoken by the learner a " " (identifying it simply as a coded line), a one-letter code for its syntactic form, and a one-letter code for pragmatic function. If, following his coding of the last utterance, the researcher subjects the file to a dictionary sort, all of the coded lines will end up together, organized alphabetically by *syntactic form*. If this researcher writes a one-line program for switching the order in which the syntactic and pragmatic codes appear on the coded line, and subjects this new data set to a dictionary sort, he can then test the other half of the contingency relationship, since he will then have the coded lines organized alphabetically by *pragmatic function*. This general procedure of arranging particular types of information in a particular order in a scratch file, and then subjecting this scratch file to a dictionary sort to reveal the structure inherent in the file, is a powerful and very generally useful tool in analysis. In addition to being created by hand coding as in the above example, scratch files may be created by extracting already-entered information and reorganizing it prior to dictionary sort.

On data coded as richly as the Individual Differences Project data, the range of possibilities for applying this reorganization and sort strategy are innumerable. Before giving examples, a more thorough description of the coding system is appropriate. In this example:

011 S: Tel' me wher'.

011 m: tell me where

011 s: (v V (n3d PN AC 1) (c (a6l WH))) #DS #IM #VC

the utterance line (S:) preserves details of what was actually said, the morphology tier (m) normalizes elements from the utterance line, eliminating upper case letters and special symbols indicating intonation, etc., and the syntax tier (s:) provides information concerning the general construction type (#DS = dative shift, #IM = imperative, #VC = verbal complement), its surface structure constituents (capital letters) inside labelled brackets, relational information ("3" in "n3d" indicates that "me" is an indirect object), and deep case information ("d" in "n3d" stands for dative). It is important to note that this rich coding scheme is intended as a descriptive device, not imputing excessive analysis to the learner, but merely characterizing the structures for purposes of helping the researcher to identify regularities of particular structures and uses.

The reorganization and sort strategy could be used for determining the distribution of the learner's utterances across flag types (i.e., the codes preceded by #) in a manner similar to that already discussed. In conjunction with this uniquely well-designed coding system, however, this strategy affords even more powerful possibilities, including a machine-generated grammar of the learner's utterances. The one-one correspondence of words on the morphology line with upper case letters on the syntax line leads easily to a dictionary of the learner's words, organized by syntactic category (e.g., all nouns, all verbs, etc.). With the use also of the bracketing from the syntax line this

reorganization and sort strategy can generate an entire dictionary of the learner's clauses, and clause types. This can be an invaluable tool in facilitating the visibility of regularities in the data.

Once the regularities are observed, it is the researcher's task to determine, for example, which of them reflect productive strategies and which of them are strictly formulaic. This task, too, is facilitated by the data organization function of computers. If the researcher observes from the listing that the only phrase in which "what" ever occurs is "what's this", he may be skeptical that "what" has any independent lexical status for the learner. Similarly, if the researcher notices from the listing that "what's this" is the only WH-question the learner ever uses, and furthermore, that it is sometimes used anomalously, this would be good evidence that this phrase is a frozen construction.

Furthermore, the data organization strategy may sometime reveal unexpected similarities across categories or unexpected subdivisions within categories which in turn lead to restructurings within certain subdomains of the coding system. Thus, these methods can even serve functions of self-modification in the course of their use with actual data.

3.4.3 Data modification: Computer-assisted coding of the data

There exist a number of high-level languages, parsers, and data base query systems which can greatly lighten the routine work of coding. Coding no longer has to be as time-consuming as previously. At the same time, it should be emphasized that the general strategies described here can perform powerful facilitating functions even in the absence of elaborate codes.

3.5 Summary and Conclusions.

In this paper, a general procedure for formatting computer data files was outlined, and several powerful analytic procedures were described. These procedures presuppose no initial coding of the data, though coding may increase the efficiency of some processes. The decision of whether or not to code data and what categories to use is largely dependent on the nature of the research question.

Computers can dramatically speed the research process if the available analytic procedures are applied resourcefully and the data are appropriately formatted. The establishment of a minimally constraining set of data formatting conventions as an international standard might help in making these procedures available to more people, and could greatly facilitate collaboration and data exchange across projects, including comparisons with first language acquisition.